Pierre-Louis NEUMANN

```
********************************
*     DEFINITION D'UN ESPACE   *
*        COMPREHENSIBLE        *
********************************
```

Mon intention dans ce programme est de définir un systeme de construction qui permette de rendre un dessin mentalement intéressant.

Il s'agit de déterminer un espace à deux dimensions pouvant être "compris" par le spectateur après qu'il ait établi les relations entre les éléments de cet espace.

Le système de construction repose sur une règle de construction qui utilise une loi géométrique visuellement identifiable (premier niveau de compréhension : repérage du processus de construction ).

Le système est initialisé par un objet de base (choisi par l'utilisateur ) qui aura une influence à toutes les étapes de la construction (deuxième niveau de compréhension : repérage du développement spécifique à chaque objet de base)

Le programme CONSTRUCTION 1 permet de construire un espace à deux dimensions à partir d'un objet quelconque, c'est à dire qu'il a pour but de placer un certain nombre d'éléments les uns par rapport aux autres en se référant à un objet de base (ou objet modèle ) situé dans le plan.

Chaque élément se présentera sous la forme d'une liste de caractéristiques. D'autre part la position de chacun d'eux sera déterminée par une loi géométrique simple. L'objet modèle sera une liste d'éléments qui devront posséder au moins les mêmes caractéristiques que ceux que l'on propose pour la construction. Cette liste s'élargira en cours de programme par adjonction de chaque nouvel élément positionné.

Le dessin final dépendra donc du choix du nombre d'éléments, de leurs caractéristiques, et de la forme de base.

Le langage LISP 510, particulièrement adapté au traitement des listes et à l'analyse des caractéristiques, sera le langage utilisé.

## DESCRIPTION DU PROGRAMME

### 1 Les éléments

Tous les éléments seront des segments de droite caractérisés par :

1/ une orientation plane : horizontale H , verticale V

2/ Une longueur quelconque L

3/ Une intensité quelconque I

Exemple : (H 5 1) ligne horizontale de longueur 5 et d'intensité 1

Les éléments de l'"objet modèle" auront comme caractéristiques supplémentaires les coordonnées de leurs extrêmités.

Exemple : (H 5 1 (6.3)(6.8))

### 2 La loi géométrique

Un élément sera positionné par rapport à un autre élément déjà positionné ; pour celà ils devront avoir même longueur, même orientation et des intensités différentes.

Soient X et Y deux éléments. Soit X déjà positionné et e,f,g,h les coordonnées de Y à calculer.

Exemple : X = (H 5 1 (a.b)(c.d))

Y = (H 5 3 e f g h)

Sachant que A = L(X)/2 les coordonnées de Y seront :

1/ si I(Y) supérieur à I(X) et orientation H ou si I(Y) inférieur à I(X) et orientation V alors e = A-a, f = A+b, g = A-c, h = A+d (Figure 1) sinon e = A+a, f = A-b, g = A+c, h = A-d (Figure 2) .

2/ si I(Y) inférieur à I(X) et même orientation H ou même orientation V alors e = A-a, f = A-b, g = A-c, h = A-d (Figure 3) sinon si I(Y) supérieur à I(X) et même orientation H ou même orientation V alors e = A+a, f = A+b, g = A+c, h = A+d (Figure 4) .

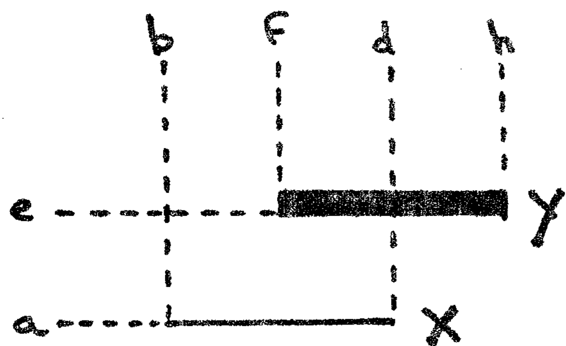
### 3 Les contraintes

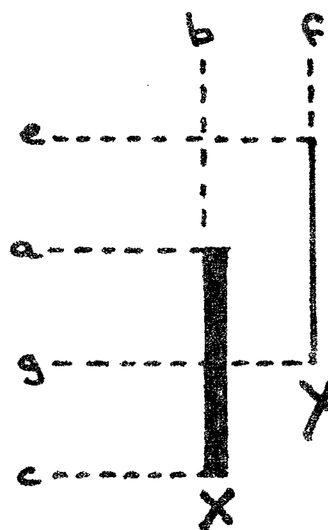
Deux éléments ne devront pas être situés sur une même ligne ou une même colonne.

Deux éléments de même intensité ne devront pas être à la fois parallèles et en regard l'un de l'autre.

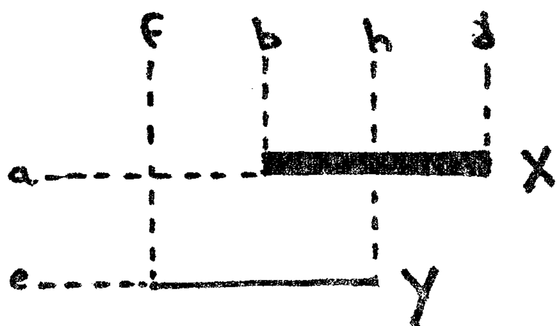
Les éléments dont la position respectera les conditions énoncées ci-dessus seront acceptés et mis en queue de la liste "objet-modèle", dans tous les autres cas ils seront refusés. Le listing indiquera les éléments refusés et acceptés, ces derniers suivis de leur position.

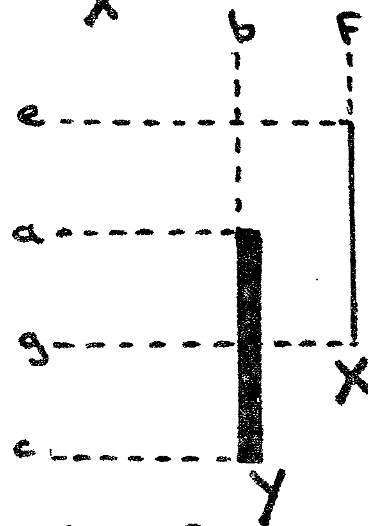
## Fig 1



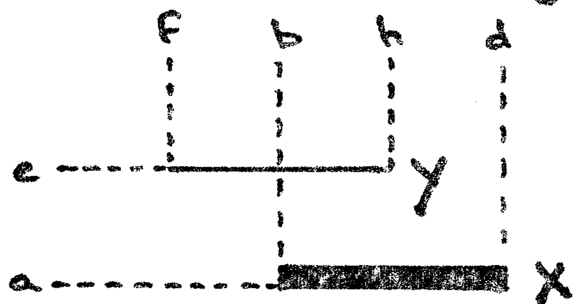



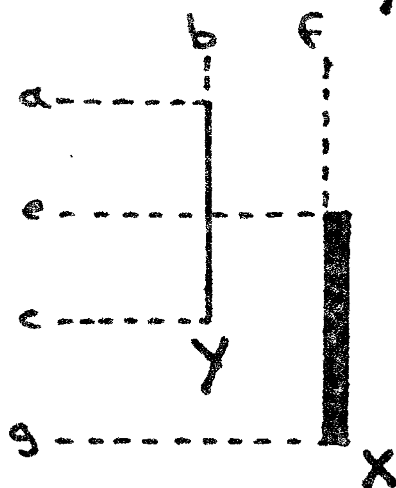

## Fig 2



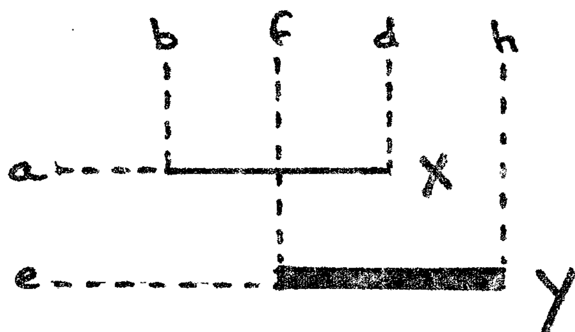



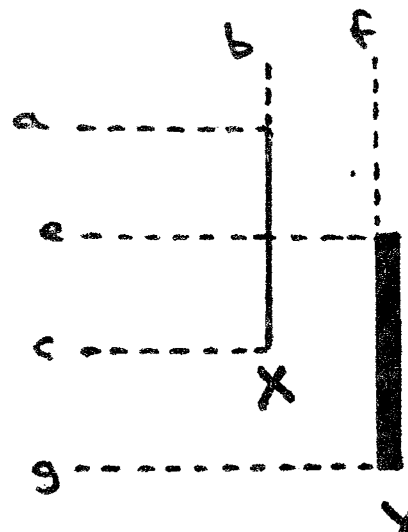

## Fig 3




## Fig 4



```
*********************
*  CONSTRUCTION 1   *
*********************
```

```
(DE CADDDR (X)
(CADDR(CDR X)))
(DE COTE1 (X)
(CADDR(CDR  X))  )
(DE COTE2 (X)
(COTE1 (CDR X)))
(DE C1A (X) (CAR(COTE1 X)))
(DE C1B (X)(CADR(COTE1 X)))
(DE C2A (X)(CAR(COTE2 X)))
(DE C2B (X) (CADR(COTE2 X)))

(DE APPEND(X Y)
(COND
((NULL X)Y)
(T(CONS(CAR X)
      (APPEND(CDR X) Y))) ))

(DE DELETE(X Y)
(COND
((NULL Y)NIL)
((EQUAL X(CAR Y))
(CDR Y))
(T(CONS(CAR Y)
(DELETE X(CDR Y)))) ))

(DE DIAG (W X Y Z)
               ;**CALCUL DES COORDONNEES**;
(LIST  (X (C1A Z) W)  (Y (C1B Z) W)
    (X (C2A Z) W)  (Y (C2B Z) W)) )

(DE NOBJET (W X Y)
               ;**AJOUTE L ELEMENT POSITIONNE EN QUEUE DE LA
                  LISTE OBJET MODEL**;
(PROGN
(PRINT(SETQ W (APPEND W (LIST
(LIST (CAR X) (CADR X))
(LIST (CADDR X) (CADDDR X))))))
(CONS(APPEND (CAR Y) (LIST W) )
  (DELETE (CADR Y) (CDR Y))) ))

(DE VERIF (V W X Y)
(COND ((EQ W (QUOTE V))
(PLACE W (QUOTE C1B) W X Y (QUOTE CADR) (QUOTE C1A)
                           (QUOTE CAR) ))
(T(PLACE W (QUOTE C1A)  W X Y (QUOTE CAR)
             (QUOTE C1B) (QUOTE CADR)))))
```

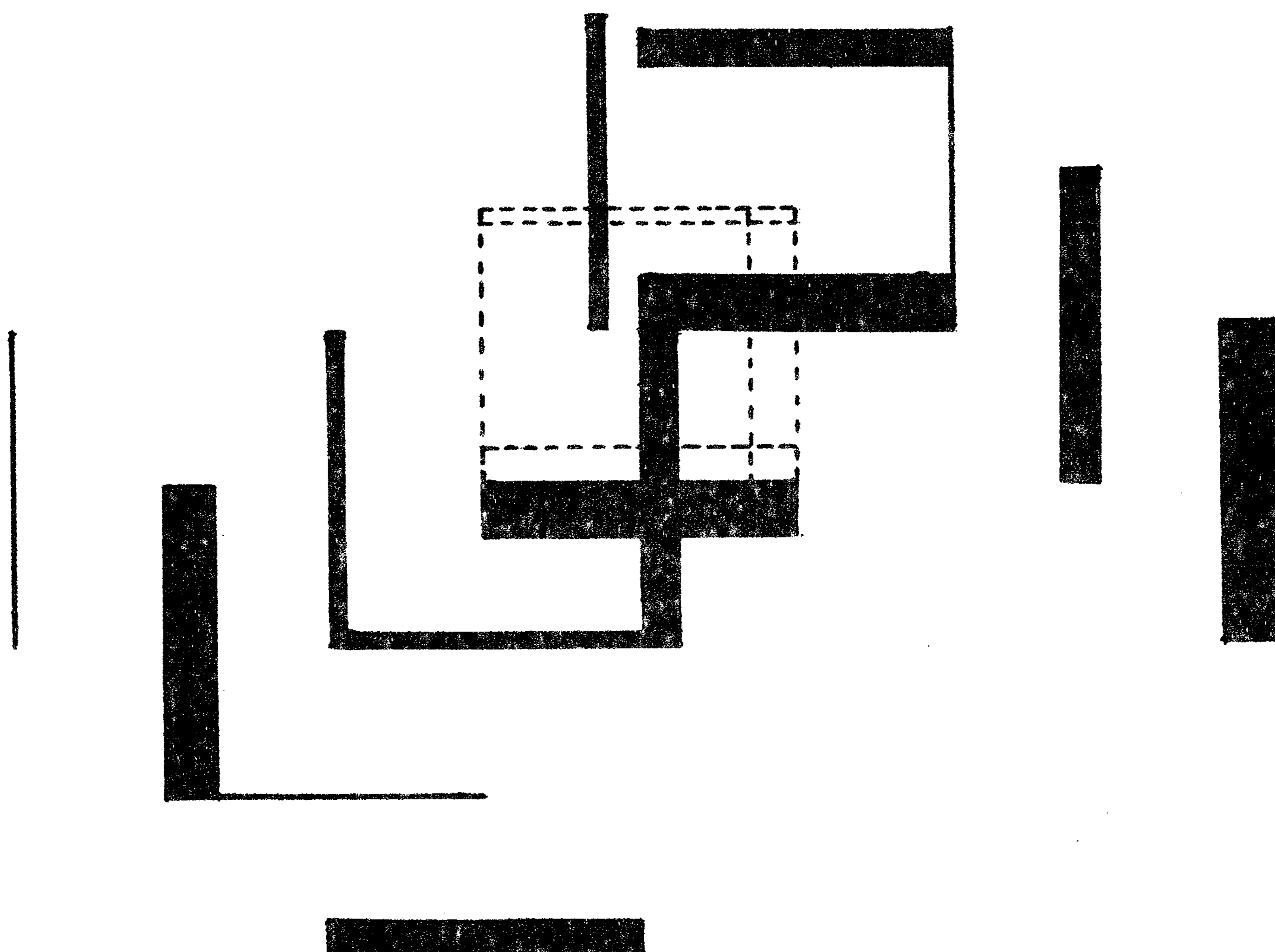
```
(DE POSIT (V X Y Z) (PROG ( O P Q R)
              ;**PROPOSE UNE POSITION DE L ELEMENT EN FONCTION
                 DE SES CARACTERISTIQUES**;
(SETQ P (CADDR Y))
(SETQ R (CADDR Z))
(SETQ O (QUO (CADR Y) 2 ))
(COND ((OR
(AND
(EQ V (QUOTE H))
(GT P R))
(AND
(EQ V (QUOTE V))
(LT P R )) )
(SETQ Q (DIAG O (QUOTE DIFFER) (QUOTE PLUS) Z )))
(T(SETQ Q (DIAG O (QUOTE PLUS) (QUOTE DIFFER) Z ))) )
(AND (VERIF Q V X P ) (RETURN Q))
(COND ((LT P R)
(SETQ Q (DIAG O (QUOTE DIFFER) (QUOTE DIFFER) Z )))
(T(SETQ Q (DIAG O (QUOTE PLUS) (QUOTE PLUS) Z ))) )
(AND (VERIF Q V X P ) (RETURN Q))
(RETURN) ))

(DE OBJET (X)(PROG(O P Q R)
              ;**COMPARE LE PREMIER ELEMENT (A POSITIONNER)
                 ET LA LISTE OBJET MODEL**;
   (SETQ O (CADR X))
(SETQ Q (CAR X))
E  (SETQ P (CAR Q))
 (COND ((NULL Q) (RETURN  ))
((AND
 (EQ (CADR P) (CADR O))
(EQ (CAR P) (CAR O))
(NOT(EQ (CADDR P) (CADDR O))))
(GO A)))
D  (SETQ Q (CDR Q))
(AND Q (GO E)) (GO F)
A  (COND ((EQ (CAR O) (QUOTE H))
(SETQ R(POSIT          (QUOTE H)       (CAR X) O P)))
(T(SETQ R(POSIT          (QUOTE V)       (CAR X) O P))))
(AND (NOT R) (GO D))
(SETQ X (NOBJET O R X))
(GO RE)
F  (PRIN1 (QUOTE REFUSE))
(PRINT O)
(SETQ X (DELETE O X))   (GO RE) ))

(DE PLACE (R S U V W X Y Z BID)
              ;**VERIFIE QUE L ELEMENT POSITIONNE
                 SATISFAIT AUX CONTRAINTES**;
(COND((NULL V) T)
((OR
(AND
(EQ (CAAR V) U)
(EQ ( S (SETQ BID (CAR V))) (X R)))
(AND
(EQ (CAAR V) U)
(EQ W (CADDR BID))
(EQ (Y BID) (Z R)) ) ))
(T(PLACE R S U (CDR V) W X Y Z BID))))
```

```
(CPUET(QUOTE
 ((
 (H 6 2 (101 101) (101 106))
 (H 6 3 (106 101) (106 106))
 (V 6 1 (101 101) (106 101))
 (V 6 4 (101 106) (106 106))
 )

 (H 6 1) (H 6 2) (H 6 3) (H 6 4)
 (H 6 1) (H 6 2) (H 6 3) (H 6 4)
 (V 6 1) (V 6 2) (V 6 3) (V 6 4)
 (V 6 1) (V 6 2) (V 6 3) (V 6 4)
 )))
( H 6 1 ( 104 98 ) ( 104 103 ) )
( H 6 2 ( 109 98 ) ( 109 103 ) )
( H 6 3 ( 98 104 ) ( 98 109 ) )
( H 6 4 ( 103 104 ) ( 103 109 ) )
( H 6 1 ( 112 95 ) ( 112 100 ) )
REFUSE ( H 6 2 )
( H 6 3 ( 115 98 ) ( 115 103 ) )
( H 6 4 ( 107 101 ) ( 107 106 ) )
( V 6 1 ( 98 109 ) ( 103 109 ) )
( V 6 2 ( 104 98 ) ( 109 98 ) )
( V 6 3 ( 104 104 ) ( 109 104 ) )
( V 6 4 ( 107 95 ) ( 112 95 ) )
( V 6 1 ( 104 92 ) ( 109 92 ) )
( V 6 2 ( 98 103 ) ( 103 103 ) )
( V 6 3 ( 101 112 ) ( 106 112 ) )
( V 6 4 ( 104 115 ) ( 109 115 ) )
NIL
```



```
(OBJET(QUOTE
 ((
 (H 6 2 (92 98) (92 103))
 (H 5 2 (99 99) (99 103))
 (H 4 2 (111 100) (111 103))
 )

(H 6 3) (H 6 4) (H 6 1) (H 6 0)
(H 5 1) (H 5 0) (H 5 4) (H 5 3)
(H 4 3) (H 4 1) (H  4 0) (H 4 4)
 )))
( H 6 3 ( 89 101 ) ( 89 106 ) )
( H 6 4 ( 95 101 ) ( 95 106 ) )
( H 6 1 ( 86 98 ) ( 86 103 ) )
( H 6 0 ( 98 98 ) ( 98 103 ) )
( H 5 1 ( 101 97 ) ( 101 101 ) )
( H 5 0 ( 97 97 ) ( 97 101 ) )
( H 5 4 ( 103 99 ) ( 103 103 ) )
( H 5 3 ( 105 97 ) ( 105 101 ) )
( H 4 3 ( 109 102 ) ( 109 105 ) )
( H 4 1 ( 107 100 ) ( 107 103 ) )
REFUSE ( H 4 0 )
( H 4 4 ( 113 102 ) ( 113 105 ) )
NIL
```

```
(CPUET(QUOTE
 ((
 ( V 9 2 (101 103) (109 103))
 (V 9 3 (101 105) (109 105))
 (H 7 2 (106 101) (106 107))
 (H 7 3 (104 101) (104 107))
 )

 (H 7 1) (H 7 2) (H 7 3)
 (H 7 1) (H 7 2) (H 7 3)
 (H 7 1) (H 7 2) (H 7 3)
 (V 9 1) (V 9 2) (V 9 3)
 (V 9 1) (V 9 2) (V 9 3)
 (V 9 1) (V 9 2) (V 9 3)
 )))
( H 7 1 ( 109 98 ) ( 109 104 ) )
( H 7 2 ( 107 98 ) ( 107 104 ) )
( H 7 3 ( 103 104 ) ( 103 110 ) )
( H 7 1 ( 110 95 ) ( 110 101 ) )
REFUSE ( H 7 2 )
( H 7 3 ( 113 98 ) ( 113 104 ) )
( H 7 1 ( 100 101 ) ( 100 107 ) )
( H 7 2 ( 116 95 ) ( 116 101 ) )
REFUSE ( H 7 3 )
( V 9 1 ( 97 107 ) ( 105 107 ) )
( V 9 2 ( 97 109 ) ( 105 109 ) )
( V 9 3 ( 105 99 ) ( 113 99 ) )
( V 9 1 ( 93 113 ) ( 101 113 ) )
REFUSE ( V 9 2 )
( V 9 3 ( 97 117 ) ( 105 117 ) )
( V 9 1 ( 101 95 ) ( 109 95 ) )
( V 9 2 ( 93 121 ) ( 101 121 ) )
REFUSE ( V 9 3 )
NIL
```

